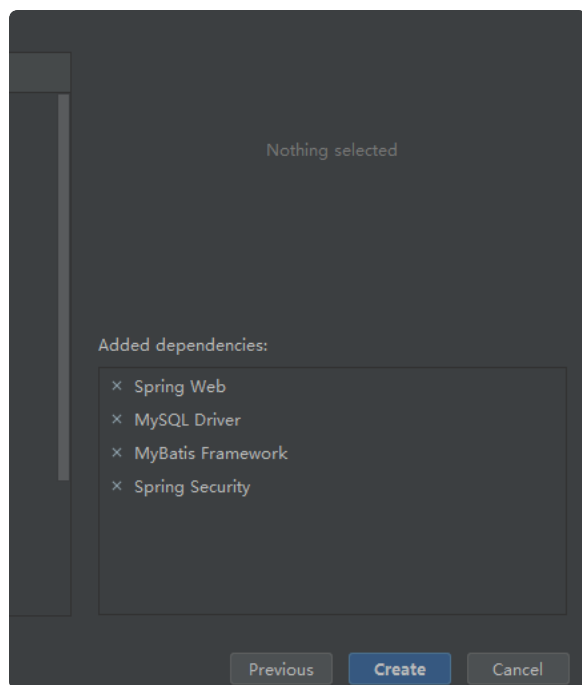


## 4.服务端环境搭建

添加以下依赖



导入数据库

 vhr.sql

用逆向工程生产controller mapper service model

项目地址:[https://github.com/lenve/generate\\_code](https://github.com/lenve/generate_code)

视频地址:<https://mp.weixin.qq.com/s/2Ovqwn1hd40D7FtKE60HKw>

生成之后添加数据库信息

```
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
spring.datasource.username=root
spring.datasource.password=111111
spring.datasource.url=jdbc:mysql:///vhr?useUnicode=true&characterEncoding=utf-8
```

配置mapper文件位置

```

package org.xiaojf.vhr;

import ...

@SpringBootApplication
@MapperScan(basePackages = "org.xiaojf.vhr.mapper")
public class VhrApplication {

    public static void main(String[] args) { SpringApplication.run(VhrApplication.class, args); }

}

```

```

</dependencies>

<build>
    <resources>
        <resource>
            <directory>src/main/resources</directory>
        </resource>
        <resource>
            <directory>src/main/java</directory>
            <includes>
                <include>/**/*.xml</include>
            </includes>
        </resource>
    </resources>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

如果发生以下错误:

```

2022-08-10 11:11:01.775 INFO 16140 --- [main] org.apache.catalina.core.StandardEngine : Starting servlet engine: [Apache Tomcat/9.0.65]
2022-08-10 11:11:01.900 INFO 16140 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-08-10 11:11:01.901 INFO 16140 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1472 ms
2022-08-10 11:11:02.027 INFO 16140 --- [main] c.a.d.s.b.a.DruidDataSourceAutoConfigure : Init DruidDataSource
2022-08-10 11:11:02.185 INFO 16140 --- [main] com.alibaba.druid.pool.DruidDataSource : {dataSource-1} inited
2022-08-10 11:11:02.362 ERROR 16140 --- [main] o.s.spring.mapper.MapperFactoryBean : Error while adding the mapper 'interface org.xiaojf.vhr.mapper.EmployeeMapper' to configuration.

org.apache.ibatis.builder.BuilderException: Error parsing Mapper XML. The XML location is 'org/xiaojf/vhr/mapper/EmployeeMapper.xml'. Cause: org.apache.ibatis.builder.BuilderException: Error resolving JDBCType.
    at org.apache.ibatis.builder.xml.XMLMapperBuilder.configurationElement(XMLMapperBuilder.java:123) ~[mybatis-3.5.9.jar:3.5.9]
    at org.apache.ibatis.builder.xml.XMLMapperBuilder.parse(XMLMapperBuilder.java:95) ~[mybatis-3.5.9.jar:3.5.9]
    at org.apache.ibatis.builder.annotation.MapperAnnotationBuilder.loadXmlResource(MapperAnnotationBuilder.java:179) ~[mybatis-3.5.9.jar:3.5.9]
    at org.apache.ibatis.builder.annotation.MapperAnnotationBuilder.parse(MapperAnnotationBuilder.java:110) ~[mybatis-3.5.9.jar:3.5.9]
    at org.apache.ibatis.binding.MapperRegistry.addMapper(MapperRegistry.java:72) ~[mybatis-3.5.9.jar:3.5.9]
    at org.apache.ibatis.session.Configuration.addMapper(Configuration.java:844) ~[mybatis-3.5.9.jar:3.5.9]
    at org.mybatis.spring.mapper.MapperFactoryBean.checkDaoConfig(MapperFactoryBean.java:80) ~[mybatis-spring-2.0.7.jar:2.0.7]
    at org.springframework.dao.support.DaoSupport.afterPropertiesSet(DaoSupport.java:44) ~[spring-tx-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.invokeInitMethods(AbstractAutowireCapableBeanFactory.java:1803) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initializeBean(AbstractAutowireCapableBeanFactory.java:1800) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:620) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:542) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractBeanFactory.lambda$doGetBean$0(AbstractBeanFactory.java:335) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:234) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:333) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:208) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.config.DependencyDescriptor.resolveCandidate(DependencyDescriptor.java:276) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanFactory.java:1391) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFactory.java:1311) ~[spring-beans-5.3.22.jar:5.3.22]
    at org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor$AutowiredFieldElement.resolveFieldValue(AutowiredAnnotationBeanPostProcessor.java:656) ~[spring-beans-5.3.22.jar:5.3.22]

```

```
ber' to configuration.  
  
Builder.BuilderException: Error resolving JdbcType. Cause: java.lang.IllegalArgumentException: No enum constant org.apache.ibatis.type.JdbcType.ENUM  
  
[3.22]  
22]
```

是因为逆向工程自动将jdbcType="ENUM" 加入了进去,但是mybatis没有这个数据类型,删除就行

```
<result column="address" property="address" jdbcType="VARCHAR" />  
<result column="departmentId" property="departmentid" jdbcType="INTEGER" />  
<result column="jobLevelId" property="joblevelid" jdbcType="INTEGER" />  
<result column="posId" property="posid" jdbcType="INTEGER" />  
<result column="engageForm" property="engageform" jdbcType="VARCHAR" />  
<result column="tiptopDegree" property="tiptopdegree" jdbcType="ENUM" />  
<result column="specialty" property="specialty" jdbcType="VARCHAR" />  
<result column="school" property="school" jdbcType="VARCHAR" />  
<result column="beginDate" property="begindate" jdbcType="DATE" />  
<result column="workState" property="workstate" />  
<result column="workID" property="workid" jdbcType="CHAR" />  
<result column="contractTerm" property="contractterm" jdbcType="DOUBLE" />  
<result column="conversionTime" property="conversiontime" jdbcType="DATE" />  
<result column="notWorkDate" property="notworkdate" jdbcType="DATE" />  
<result column="beginContract" property="begincontract" jdbcType="DATE" />  
<result column="endContract" property="endcontract" jdbcType="DATE" />  
<result column="workAge" property="workage" jdbcType="INTEGER" />
```

MyBatis包含的JdbcType类型，主要有下面这些，大致了解一下即可：

BIT、FLOAT、CHAR、TIMESTAMP、OTHER、UNDEFINEDTINYINT、REAL、VARCHAR、  
BINARY、BLOB NVARCHAR、SMALLINT、DOUBLE、LONGVARCHAR、VARBINARY、CLOB、  
NCHAR、INTEGER、NUMERIC、DATE、LONGVARBINARY、BOOLEAN、NCLOB、BIGINT、  
DECIMAL、TIME、NULL、CURSOR

上述JdbcType类型和Java类型的对应关系，可以参照下面的列表，不过不同数据库的JdbcType多少有些出入，请注意即可。

1	JDBC Type	Java Type
2	CHAR	String
3	VARCHAR	String
4	LONGVARCHAR	String
5	NUMERIC	java.math.BigDecimal
6	DECIMAL	java.math.BigDecimal
7	BIT	boolean
8	BOOLEAN	boolean
9	TINYINT	byte
10	SMALLINT	short
11	INTEGER	INTEGER
12	BIGINT	long
13	REAL	float
14	FLOAT	double
15	DOUBLE	double
16	BINARY	byte[]
17	VARBINARY	byte[]
18	LONGVARBINARY	byte[]
19	DATE	java.sql.Date
20	TIME	java.sql.Time
21	TIMESTAMP	java.sql.Timestamp
22	CLOB	Clob
23	BLOB	Blob
24	ARRAY	Array
25	DISTINCT	mapping of underlying type
26	STRUCT	Struct
27	REF	Ref
28	DATALINK	java.net.URL